Neighborhood of Actors, Who is the MVP among them?

Team 8 - Network Tour of Data Science

1. Story

The idea of the project is to find communities of actors, explain how they are related within their communities and find the most representative actors.

We want to understand the relationships between actors because there might be implicit groups of actors that perform on similar movies, interact with the same people, work for the same productions companies, etc. Within these communities we want to find the most representative actors and what genre of movies they have taken part on most frequently. So with this information, a new actor can know which type of movies he should perform on in order to eventually become part of this community.

The IMDb dataset is relevant for this work because it provides reliable information about the movies each actor has performed on, the people he/she has interacted with, the production companies he/she has worked for, etc. We can use this information to infer affinity between actors by looking at how many of these elements they have in common. The dataset can be downloaded from kaggle : https://www.kaggle.com/tmdb/tmdb-movie-metadata.

We are building a graph from the data and analyzing it with the following specialized tools: Pandas, Scikit-learn, Networkx, Matplotlib, Python louvain.

2. Acquisition

We used the TMDB 5000 Movie dataset provided by Kaggle.

- **Description of the data**
  The dataset consisted of two tables containing features from movies. The first table provided the following information:
  - movie id: movie unique identifier.
  - title
  - cast: a column of json strings containing actor names, order (importance, the value is 0 for the main actor), gender, id and character.
  - crew: a column of json strings with the name, department, id , gender and job of the crew members. The second table contained the following columns:
  - budget
  - genres

- homepage
- movie_id
- keywords
- original_language
- original_title
- overview
- popularity
- production_companies
- production_countries
- release_date
- revenue
- runtime
- spoken_languages
- status
- tagline
- title
- vote_average
- vote_count

- **Creation of the graph**

  As we wanted to create a graph of actors we had to process the table in the following way:

  - We extracted the actors from the cast column of the first table transforming it into a table of actors. Since there were around 54000 (be more precise) entries in total we took into account only the protagonist (actors with order 0) reducing the table to 2000 entries (be more precise).
  - We joined the actors table with the second table and, for each actor, we aggregated the features of the movies as follows:
    - movie_id: set of movie ids
    - cast: union of casts

- crew: union of crews
- actor_id: max actor id (since the value is the same across movies the max operator does not have any effect)
- gender: max gender (same argument as for actor_id)
- budget: mean of budgets across movies
- genres: union of genres
- keywords: union of keywords
- original_language : set of original languages
- popularity: mean popularity across movies
- production_companies: union
- production_countries: union
- release_date: list
- revenue: mean of revenues across movies
- runtime: sum
- spoken_languages: union
- status: list
- title: set
- vote_average: mean
- vote_count: mean (Should be sum?)

– We defined the "affinity" (weights) between two actors by the following formula:

$$w_{ij} = \frac{0.3|movie\_id_i \cap movie\_id_j| + 0.3|cast_i \cap cast_j| + 0.2|crew_i \cap crew_j| + 0.1|genre_i \cap genre_j}{0.3|movie\_id_i \cup movie\_id_j| + 0.3|cast_i \cup cast_j| + 0.2|crew_i \cup crew_j| + 0.1|genre_i \cup genre_j}$$

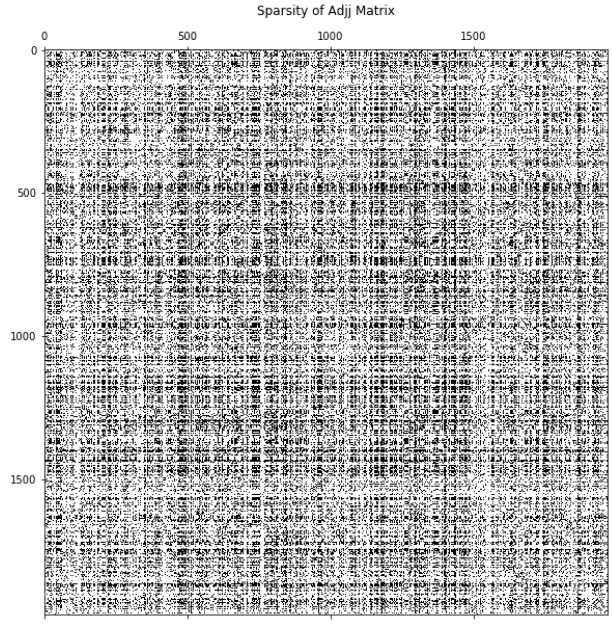
– This means that actors that share most of their cast, movies, genres and production companies are strongly related.

3. Exploration

The first exploration that we did show the following:

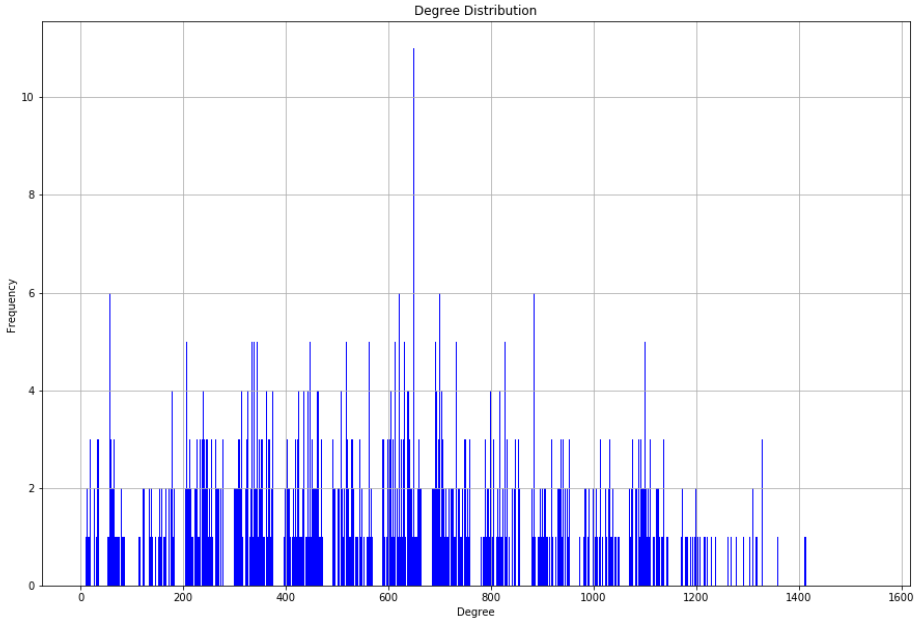
- **Connected components:**
  Number of connected components 2, however we decided to sparsifiy more the matrix so that is manageable and feasible to use, obtained one connected component

- **Sparsity of the graph:**

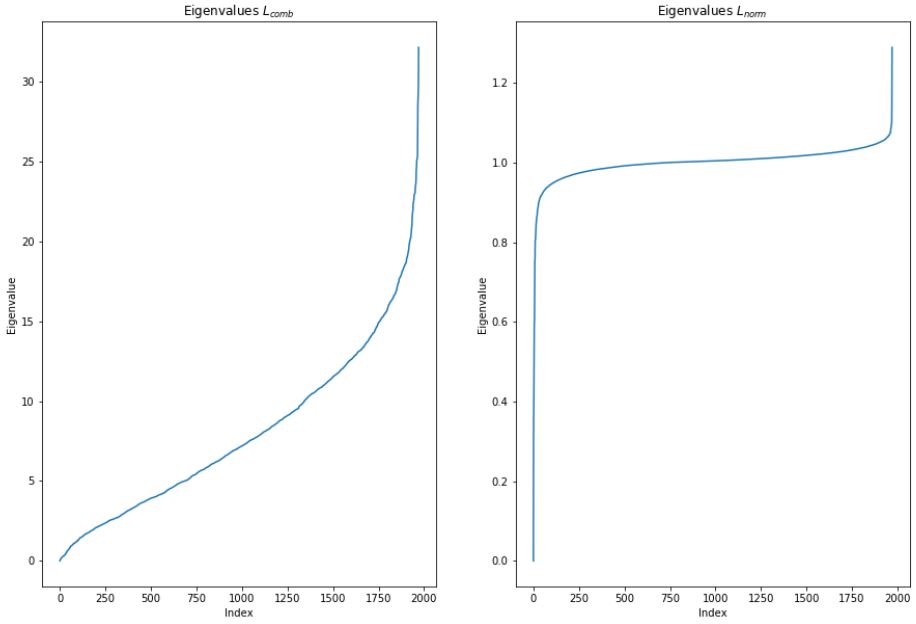


- **Diameter:**
  4, meaning that any actor is 4 steps away of knowing any other actor.

- **Degree distribution:**



- **Spectrum:**

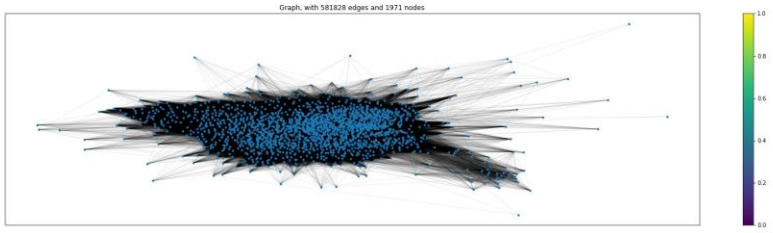


- **Type of graph:**
  The network is a small world, to get this assumption a similar generated Erdős–Rényi network was created and network statistics like clustering coefficient and the mean shortest path were found for both. Mean shortest path is the same, however, clustering coefficient is different.
  Small networks should have some spatial structure, that is reflected on a bigger clustering coefficient.

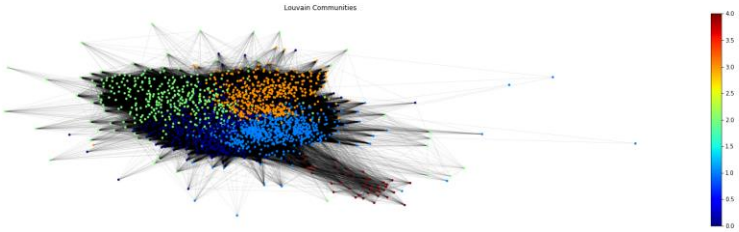
- **Properties of the nodes:**
  - Average Degree: 590.388635210553
  - Average clustering coefficient: 0.6403268534042411
  - Nodes with higher centrality:
    - Britney Spears - 0.7812182741116752
    - Melissa Joan Hart - 0.7578680203045686
    - Orlando Jones - 0.733502538071066
    - Nicholas Rowe - 0.7324873096446701
    - Tom Selleck - 0.7279187817258883
  - Nodes with small centrality:
    - Ben Youcef 0.0040609137055837565
    - Caitlin Fitzgerald 0.003553299492385787
    - Kirby Heyborne 0.0015228426395939086
    - J.D. Williams 0.0010152284263959391
    - Toshirō Mifune 0.0010152284263959391
  - Hub Nodes (Just some actors, because number of hubs is 964):
    - 50 Cent
    - AJ Michalka
    - Aaron Abrams
    - Aaron Stanford
    - Aasheekaa Bathija
    - Zhang Ziyi
    - Zoe Kazan

- Zoe Lister-Jones
- Zoe Saldana
- Zooey Deschanel

- **Analysis of the attributes:**
- **A visualization of the network:**



4. Exploitation



- We used Louvain's algorithm to find the communities of the graph, this algorithm is a bottom up approach to find communities based on the modularity of the nodes. The idea is to use these values a ground-truth so that later we can train a Machine Learning model (Logistic Regression) so that we try to recreate these results based on the features of the actors

5. Next steps

- Analyze the communities and try to find the relationships that explain them.
- Find most representative actors of each community.
- Create appropriate visualizations.
- Logistic Regression on the labeled data (louvain graph) to try to find the features that are the most relevant in the community formation and have a Machine Learning model able to predict to which community an actor would belong.